

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Inventors:	Antti Kokkinen, et al.	Examiner:	Ben C. Wang
Serial No.:	10/688,640	Group Art Unit:	2192
Filed:	October 17, 2003	Docket No.:	200701903-2
Title:	UPDATE SYSTEM EMPLOYING REFERENCE SOFTWARE TO REDUCE NUMBER OF UPDATE PACKAGES		

---

**APPEAL BRIEF UNDER 37 C.F.R. § 41.37**

Mail Stop Appeal Brief - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

This Appeal Brief is filed in response to the Final Office Action mailed August 6, 2010 and Notice of Appeal mailed November 6, 2010.

**AUTHORIZATION TO DEBIT ACCOUNT**

It is believed that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required (including fees for net addition of claims) are hereby authorized to be charged to Hewlett-Packard Development Company's deposit account no. 08-2025.

### **I. REAL PARTY IN INTEREST**

The real party in interest is Hewlett-Packard Development Company, LP, a limited partnership established under the laws of the State of Texas and having a principal place of business at 11445 Compaq Center Drive West, Houston, TX 77070, U.S.A. (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC.

## **II. RELATED APPEALS AND INTERFERENCES**

There are no known related appeals or interferences known to Appellant, Appellant's legal representative, or assignee that will directly affect or be directly affected by or have a bearing on the Appeal Board's decision in the pending appeal.

### **III. STATUS OF CLAIMS**

Claims 1 – 24 are pending in the application and stand finally rejected.  
The rejection of claims 1 – 24 is appealed.

#### **IV. STATUS OF AMENDMENTS**

No amendments were made after receipt of the Final Office Action. All amendments have been entered.

## **V. SUMMARY OF CLAIMED SUBJECT MATTER**

The following provides a concise explanation of the subject matter defined in each of the independent claims involved in the appeal, referring to the specification by page and line number and to the drawings by reference characters, as required by 37 C.F.R. § 41.37(c)(1)(v). Each element of the claims is identified by a corresponding reference to the specification and drawings where applicable. Note that the citation to passages in the specification and drawings for each claim element does not imply that the limitations from the specification and drawings should be read into the corresponding claim element or that these are the sole sources in the specification supporting the claim features.

### **Claim 1**

A method for updating software in an electronic device, the method comprising:

generating an update package for updating at least one software application, the update package being generated based upon difference information between the at least one software application and at least one reference software installed on the electronic device; (see lines 2-4 of paragraph [0054] on p. 13; lines 3-9 of paragraph [0055] on p. 13)

updating the at least one software application using the update package and the at least one reference software; (Figure 2 at block 223; see lines 5-8 of paragraph [0048] on pages 11-12)

wherein the updating leaves the at least one reference software unchanged and the at least one reference software includes files common to a plurality of versions of the software application. (see lines 8-12 of paragraph [0040] on p. 9)

Claim 12

A system for updating software, the system comprising: (Figure 1 at 105)  
an electronic device (Figure 1 at 107) capable of having software (Figure 1 at 125) installed thereon; (see lines 1-2 of paragraph [0039] on p. 9)

a software delivery device (Figure 1 at 123) for receiving and installing a reference software (Figure 1 at 123) to the electronic device if the electronic device does not have the reference software previously installed; (see lines 3-5 of paragraph [0041] on p. 10)

the software delivery device receiving and delivering at least one update package to the electronic device, wherein the at least one update package is based on differences between at least one application software and the reference software, and the reference software facilitates, using the at least one update package, at least one update to the application software installed on the electronic device, and wherein the updating leaves the reference software unchanged, and the reference software includes a plurality of shared files with the application software. (see lines 8-12 of paragraph [0040] on p. 9; lines 3-5 of paragraph [0054] on p. 12)

Claim 21

A method for updating software in an electronic device, the method comprising:

generating a first update package for updating at least one software application, the first update package being generated based upon difference information between first and second software versions; (see Figure 3 and lines 3-5 of paragraph [0055] on p. 13)

generating a second update package for updating the at least one software application, the second update package being generated based upon difference information between first and third software versions; (see Figure 3 and lines 5-6 of paragraph [0055] on p. 13)

generating a third update package for updating the at least one software application, the third update package being generated based upon difference

information between the first and second update packages; (see Figure 3 and lines 7-9 of paragraph [0055] on p. 13)

updating the at least one software application using the third update package. (see Figure 3 and lines 1-2 of paragraph [0057] on p. 13)

Claim 22

A method for updating software in an electronic device, the method comprising:

generating a first update package for updating at least one software application, the first update package being generated based upon difference information between a first software version and a reference software corresponding to the at least one software application; (see Figure 3 and lines 3-5 of paragraph [0055] on p. 13)

generating a second update package for updating the at least one software application, the second update package being generated based upon difference information between a second software version and the reference software corresponding to the at least one software application; (see Figure 3 and lines 5-6 of paragraph [0055] on p. 13)

generating a third update package for updating the at least one software application, the third update package being generated based upon difference information between the first and second update packages; (see Figure 3 and lines 7-9 of paragraph [0055] on p. 13)

updating the at least one software application using the third update package, wherein the reference software includes files common to the at least one software application and to the second software version. (see Figure 3 and lines 1-2 of paragraph [0057] on p. 13; lines 8-12 of paragraph [0040] on p. 9)

Claim 24

A system for updating software, the system comprising: (Figure 1 at 105)  
an electronic device (Figure 1 at 107) capable of having software (Figure 1 at 125) installed thereon; (see lines 1-2 of paragraph [0039] on p. 9)



a first update package generator (Figure 3 at 370) for generating update packages based upon difference information between a version of software and a reference software (Figure 1 at 123) corresponding to at least one software application; (see Figure 3 and lines 3-5 of paragraph [0055] on p. 13)

a second update package generator (Figure 3 at 380) for generating update packages based upon difference information between different update packages; (see Figure 3 and lines 5-6 of paragraph [0055] on p. 13)

a software delivery device (Figure 1 at 123) for delivering at least one update package generated based upon difference information between different update packages to the electronic device, wherein the reference software includes at least one of a plurality of shared binaries, firmware code, dynamic link libraries (DLLs), and JAVA archives (JAR files) with the at least one software application. (see lines 3-5 of paragraph [0041] on p. 10; see lines 8-12 of paragraph [0040] on p. 9; lines 3-5 of paragraph [0054] on p. 12)

**VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL**

Claims 1 – 24 are rejected under 35 USC § 103(a) as being unpatentable over USPN 6,999,976 (Abdallah) in view of US publication number 2004/0062130 (Chiang).

## **VII. ARGUMENT**

The rejection of claims 1 – 24 is improper, and Appellants respectfully request reversal of these rejections.

The claims do not stand or fall together. Instead, Appellants present separate arguments for various independent and dependent claims. Each of these arguments is separately argued below and presented with separate headings and sub-heading as required by 37 C.F.R. § 41.37(c)(1)(vii).

### **Principles of Law: Obviousness**

The test for determining if a claim is rendered obvious by one or more references for purposes of a rejection under 35 U.S.C. § 103 is set forth in *KSR International Co. v. Teleflex Inc.*, 550 U.S. 398, 82 USPQ2d 1385 (2007):

Under §103, the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background the obviousness or nonobviousness of the subject matter is determined. Such secondary considerations as commercial success, long felt but unsolved needs, failure of others, etc., might be utilized to give light to the circumstances surrounding the origin of the subject matter sought to be patented. Quoting *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1 (1966).

As set forth in MPEP 2143.03, to ascertain the differences between the prior art and the claims at issue, “[a]ll claim limitations must be considered” because “all words in a claim must be considered in judging the patentability of that claim against the prior art.” *In re Wilson*, 424 F.2d 1382, 1385.

According to the Examination Guidelines for Determining Obviousness Under 35 U.S.C. 103 in view of *KSR International Co. v. Teleflex Inc.*, Federal

Register, Vol. 72, No. 195, 57526, 57529 (October 10, 2007), once the *Graham* factual inquiries are resolved, there must be a determination of whether the claimed invention would have been obvious to one of ordinary skill in the art based on any one of the following proper rationales:

(A) Combining prior art elements according to known methods to yield predictable results; (B) Simple substitution of one known element for another to obtain predictable results; (C) Use of known technique to improve similar devices (methods, or products) in the same way; (D) Applying a known technique to a known device (method, or product) ready for improvement to yield predictable results; (E) “Obvious to try”—choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; (F) Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations would have been predictable to one of ordinary skill in the art; (G) Some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention. *KSR International Co. v. Teleflex Inc.*, 550 U.S. 398, 82 USPQ2d 1385 (2007).

Furthermore, as set forth in *KSR International Co. v. Teleflex Inc.*, quoting from *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006), “[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.”

Therefore, if the above-identified criteria and rationales are not met, then the cited reference(s) fails to render obvious the claimed invention and, thus, the claimed invention is distinguishable over the cited reference(s).

**Claim Rejections: 35 USC § 103(a)**

Claims 1 – 24 are rejected under 35 USC § 103(a) as being unpatentable over USPN 6,999,976 (Abdallah) in view of US publication number 2004/0062130 (Chiang). These rejections are traversed.

Differences Between the Art and Claims

Claims 1 – 24 recite one or more elements that are not taught or suggested in Abdallah in view of Chiang. These missing elements show that the differences between the combined teachings in the art and the recitations in the claims are great. As such, the pending claims are not obvious in view of or a predictable variation of the art to one of ordinary skill in the art.

These differences are shown below and presented with separate headings for different claim groups.

Sub-Heading: Claims 1 – 20

Independent claim 1 is selected for discussion.

As one example, claim 1 recites generating an update package based upon difference information between the at least one software application and at least one reference software installed on the electronic device. The examiner argues that these recitations are taught in Abdallah at column 6, lines 9-17 and 36-48 and column 5, lines 30-41. Appellants respectfully traverse.

Column 6, lines 9-17 and 30-41 in Abdallah teaches an initial file system tree that is a snapshot at a point in time of an application installation, and a final file system tree that is a snapshot at another point in time of the application installation. A delta class computes a difference between the initial and final file system trees.

Column 5, lines 30-41 in Abdallah teaches that two file system trees are compared, and the resulting difference information is encoded into a Java archive.

The cited sections of Abdallah teach a traditional method to install updates wherein a difference is obtained between an application already installed on an

electronic device and an updated application having revisions. Some shortcomings of this traditional method are discussed in paragraph [0007] of Appellants' background.

The cited sections of Abdallah fail to teach or suggest updating software installed on an electronic device using reference software also installed on the electronic device. Claim 1 recites generating an update package based upon difference information between the software application and at least one reference software installed on the electronic device. **Abdallah is missing the recitations of the reference software as recited in claim 1.**

Appellants respectfully ask the BPAI to interpret the recitations of claim 1 consistent with the specification. With regard to reference software, Appellants' specification states as follows (portions omitted for brevity):

[0040] In an embodiment of the present invention, when a user of electronic device 107 installs application software 125, for example for the first time, reference software 123 may also be simultaneously installed to support subsequent updates to application software 125.

[0042] Aspects of the present invention may utilize a known reference, such as reference software 123, when updating available software in an electronic device... Subsequent updates (for example to version 5.0) applied to the installed software may then be performed based on reference software 123 instead of application software 125 currently in use. This means that for a given updated application software version, only one update package based on reference software 123 may need to be developed and disseminated.

[0043] Aspects of the present invention may decrease the need for creating and developing numerous redundant update packages.

Multi-version updates may be combined and converted to a single update situation. For example, if application software versions 1, 2, and 3 exist, distributing an updated version of the software, for example version 4, may have required update packages to be generated from versions 1 to 4, 2 to 4, and 3 to 4 using prior art methods. However, by employing an embodiment of the present invention, only one update package may need to be developed, i.e., one update package is capable of supporting the transition from an existing reference software version to updated software version 4.

Abdallah teaches installing updates based on a difference between an application already installed on an electronic device and an updated application to be installed on the electronic device. This difference, however, is not based on the software application and reference software installed on the electronic device as recited in claim 1.

For at least these reasons, claims 1 – 20 are allowable over Abdallah in view of Chiang.

As another example, claim 1 recites that the updating leaves the reference software unchanged. The examiner argues that this claim element is taught in Figure 4 and column 5, lines 41-54. Appellants respectfully traverse.

Figure 4 and column 5, lines 41-54 in Abdallah teach an initial file system and a final file system. Abdallah computes difference information between these two file systems and stores this information. **Nowhere does Abdallah teach or even suggest that the updating leaves the reference software installed on the electronic device unchanged.** Again, this section of Abdallah merely teaches computing a difference between an initial file system and a final file system.

For at least these reasons, claims 1 – 20 are allowable over Abdallah in view of Chiang.

Sub-Heading: Claim 3

Dependent claim 3 recites the update package is performed based on the reference software installed on the electronic device instead of the software package such that only one update package is disseminated to update multiple different versions of the software application. The examiner argues that these recitations are taught in column 4, lines 61-65 of Abdallah. Appellants respectfully traverse.

Column 4, lines 61-65 of Abdallah teaches that Java applications are distributed using a JAR file. "A JAR file contains all the resources required to install and run a Java program in a single file...." Nowhere does this section teach or even suggest that the update package is performed based on the reference software installed on the electronic device instead of the software package. By contrast, Abdallah teaches that the update package is performed based on difference information between the application installed on the electronic device and the update to be installed on the electronic device.

Furthermore, Abdallah fails to teach or suggest that only one update package is disseminated to update multiple different versions of the software application. Instead, Abdallah teaches that a JAR file has all the resources required to install and run a Java program in a single file.

For at least these reasons, claim 3 is allowable over Abdallah in view of Chiang.

Sub-Heading: Claim 4

Dependent claim 4 recites updating multiple update versions of the at least one software application installed on the electronic device is performed using a single update package. The examiner argues that these recitations are taught in column 4, lines 61-65 of Abdallah. Appellants respectfully traverse.

Column 4, lines 61-65 of Abdallah teaches that Java applications are distributed using a JAR file. "A JAR file contains all the resources required to install and run a Java program in a single file...." Nowhere does this section teach or even suggest updating multiple update versions of the at least one



software application installed on the electronic device is performed using a single update package.

For at least these reasons, claim 4 is allowable over Abdallah in view of Chiang.

Sub-Heading: Claim 5

Dependent claim 5 recites installing the at least one software application and the at least one reference software as part of a single installation. The examiner argues that these recitations are taught in column 4, lines 61-65 of Abdallah. Appellants respectfully traverse.

Column 4, lines 61-65 of Abdallah teaches that Java applications are distributed using a JAR file. "A JAR file contains all the resources required to install and run a Java program in a single file...." Nowhere does this section teach or even suggest installing the at least one software application and the at least one reference software as part of a single installation. As explained in connection with independent claim 1, Abdallah does not teach or suggest this reference software.

For at least these reasons, claim 5 is allowable over Abdallah in view of Chiang.

Sub-Heading: Claim 6

Dependent claim 6 recites updating the at least one reference software and updating the at least one software application as part of a single update. The examiner argues that these recitations are taught in column 4, lines 61-65 of Abdallah. Appellants respectfully traverse.

Column 4, lines 61-65 of Abdallah teaches that Java applications are distributed using a JAR file. "A JAR file contains all the resources required to install and run a Java program in a single file...." Nowhere does this section teach or even suggest updating the at least one reference software and updating the at least one software application as part of a single update. As explained in

connection with independent claim 1, Abdallah does not teach or suggest this reference software.

For at least these reasons, claim 6 is allowable over Abdallah in view of Chiang.

Sub-Heading: Claims 21-24

Independent claim 21 is selected for discussion.

Independent claim 21 recites generating a third update package for updating the at least one software application, the third update package being generated based upon difference information between the first and second update packages. The examiner argues that column 1, lines 7-12 of Abdallah. Appellants respectfully traverse.

Column 1, lines 7-12 of Abdallah makes a general statement that a Java archive can be used to encode file system update information. The examiner has failed to establish a prima facie case to reject claim 21 based on this citation to Abdallah. This section of Abdallah does not teach or even suggest generating a third update package for updating the at least one software application, the third update package being generated based upon difference information between the first and second update packages.

Furthermore, claim 21 recites “generating a first update package for updating at least one software application, the first update package being generated based upon difference information between first and second software versions.” The claim further recites “generating a second update package for updating the at least one software application, the second update package being generated based upon difference information between first and third software versions.” Notice that the third update package is based on difference information between these first and second update packages. By contrast, Abdallah teaches installing updates based on a difference between an application already installed on an electronic device and an updated application to be installed on the electronic device.

For at least these reasons, claims 21-24 are allowable over Abdallah in view of Chiang.

### **CONCLUSION**

In view of the above, Appellants respectfully request the Board of Appeals to reverse the Examiner's rejection of all pending claims.

Any inquiry regarding this Amendment and Response should be directed to Philip S. Lyren at Telephone No. 832-236-5529. In addition, all correspondence should continue to be directed to the following address:

**Hewlett-Packard Company**  
Intellectual Property Administration  
3404 E. Harmony Road  
Mail Stop 35  
Fort Collins, CO 80528

Respectfully submitted,

/Philip S. Lyren #40,709/

Philip S. Lyren  
Reg. No. 40,709  
Ph: 832-236-5529

### **VIII. Claims Appendix**

1. A method for updating software in an electronic device, the method comprising:

generating an update package for updating at least one software application, the update package being generated based upon difference information between the at least one software application and at least one reference software installed on the electronic device;

updating the at least one software application using the update package and the at least one reference software; and

wherein the updating leaves the at least one reference software unchanged and the at least one reference software includes files common to a plurality of versions of the software application.

2. The method according to claim 1, wherein the files common to a plurality of versions of the software application includes at least one of binaries, firmware code, dynamic link libraries (DLLs), and JAVA archives (JAR files).

3. The method according to claim 1, wherein the update package is performed based on the reference software installed on the electronic device instead of the software package such that only one update package is disseminated to update multiple different versions of the software application.

4. The method according to claim 1, further comprising updating multiple update versions of the at least one software application installed on the electronic device is performed using a single update package.

5. The method according to claim 1, further comprising installing the at least one software application and the at least one reference software as part of a single installation.

6. The method according to claim 1, further comprising updating the at least one reference software and updating the at least one software application as part of a single update.

7. The method according to claim 1, wherein the at least one software application comprises a plurality of software applications, and the at least one reference software comprises a plurality of reference software.

8. The method according to claim 7, further comprising:  
identifying a software application needing updating from the plurality of software applications installed on the electronic device; and  
identifying whether a reference software corresponding to the software application needing updating is present on the electronic device, wherein if the reference software is not present, then installing the software application and an associated reference software in a single update on the electronic device.

9. The method according to claim 7, further comprising:
- identifying a software application needing updating from the plurality of software applications installed on the electronic device; and
  - identifying whether a reference software corresponding to the software application needing updating is present on the electronic device, wherein if the reference software is present, then retrieving an update package for the software application needing updating;
  - verifying the update package; and
  - installing the update package on the electronic device.
10. The method according to claim 7, further comprising:
- identifying a software application needing updating from the plurality of software applications installed on the electronic device;
  - determining if the update is needed immediately; and
  - storing the update until the update is needed immediately.
11. The method according to claim 10, wherein when the update is determined to be needed immediately, then
- invoking an update agent to employ at least the stored update package and reference software; and
  - updating the software application with the update package.

12. A system for updating software, the system comprising:
- an electronic device capable of having software installed thereon;
  - a software delivery device for receiving and installing a reference software to the electronic device if the electronic device does not have the reference software previously installed; and
  - the software delivery device receiving and delivering at least one update package to the electronic device, wherein the at least one update package is based on differences between at least one application software and the reference software, and the reference software facilitates, using the at least one update package, at least one update to the application software installed on the electronic device, and wherein the updating leaves the reference software unchanged, and the reference software includes a plurality of shared files with the application software.
13. The system according to claim 12, wherein the plurality of shared files include binaries and firmware code.
14. The system according to claim 12, further comprising an update generating system, the update generating system comprising a loader manager, the loader manager:
- managing loading of application software and application software version updates from the software delivery device;
  - employing a loader from a loader module; and

employing security services to authenticate software being delivered.

15. The system according to claim 12, wherein the plurality of shared files include dynamic link libraries (DLLs) and JAVA archives (JAR files) loader manager further comprises an installation agent for installing application software and downloading files from the software delivery device.

16. The system according to claim 14, wherein the loader manager is adapted to:

identify an application software needing updating;

identify whether reference software associated with the application software needing updating exists; and

coordinating an update of the application software and an associated reference software in a single update.

17. The system according to claim 14, wherein the loader manager is adapted to:

retrieve the update package;

access contents of the update package; and

verify the update package.

18. The system according to claim 14, wherein the loader manager is adapted to determine immediacy of a needed update for a particular application software.



19. The system according to claim 12, wherein the software delivery device is one of a server, a CDROM, and a network.

20. The system according to claim 12, wherein the electronic device is one of a computer, a digital phone, and a digital camera.

21. A method for updating software in an electronic device, the method comprising:

generating a first update package for updating at least one software application, the first update package being generated based upon difference information between first and second software versions;

generating a second update package for updating the at least one software application, the second update package being generated based upon difference information between first and third software versions;

generating a third update package for updating the at least one software application, the third update package being generated based upon difference information between the first and second update packages; and

updating the at least one software application using the third update package.

22. A method for updating software in an electronic device, the method comprising:

generating a first update package for updating at least one software application, the first update package being generated based upon difference information between a first software version and a reference software corresponding to the at least one software application;

generating a second update package for updating the at least one software application, the second update package being generated based upon difference information between a second software version and the reference software corresponding to the at least one software application;

generating a third update package for updating the at least one software application, the third update package being generated based upon difference information between the first and second update packages; and

updating the at least one software application using the third update package, wherein the reference software includes files common to the at least one software application and to the second software version.

23. The method of claim 22, wherein the files include at least one of binaries, firmware code, dynamic link libraries (DLLs), and JAVA archives (JAR files).

24. A system for updating software, the system comprising:

an electronic device capable of having software installed thereon;

a first update package generator for generating update packages based upon difference information between a version of software and a reference software corresponding to at least one software application;

a second update package generator for generating update packages based upon difference information between different update packages; and

a software delivery device for delivering at least one update package generated based upon difference information between different update packages to the electronic device, wherein the reference software includes at least one of a plurality of shared binaries, firmware code, dynamic link libraries (DLLs), and JAVA archives (JAR files) with the at least one software application.

**IX. EVIDENCE APPENDIX**

None.

**X. RELATED PROCEEDINGS APPENDIX**

None.